Biclustering Algorithms for Gene Expression Analysis

T. M. Murali

August 19, 2008

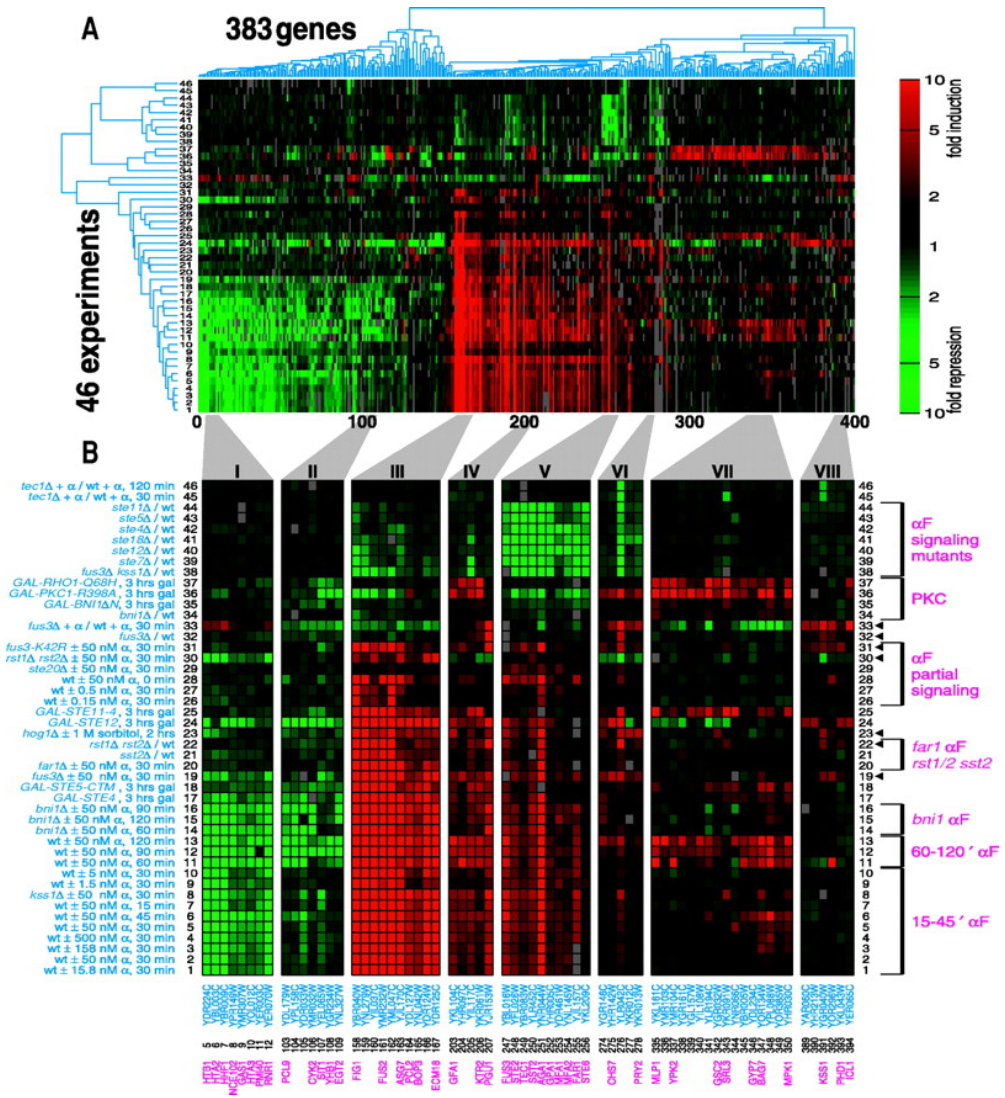
Problems with Hierarchical Clustering

- It is a global clustering algorithm.
- Considers *all* genes to be equally important for all samples.

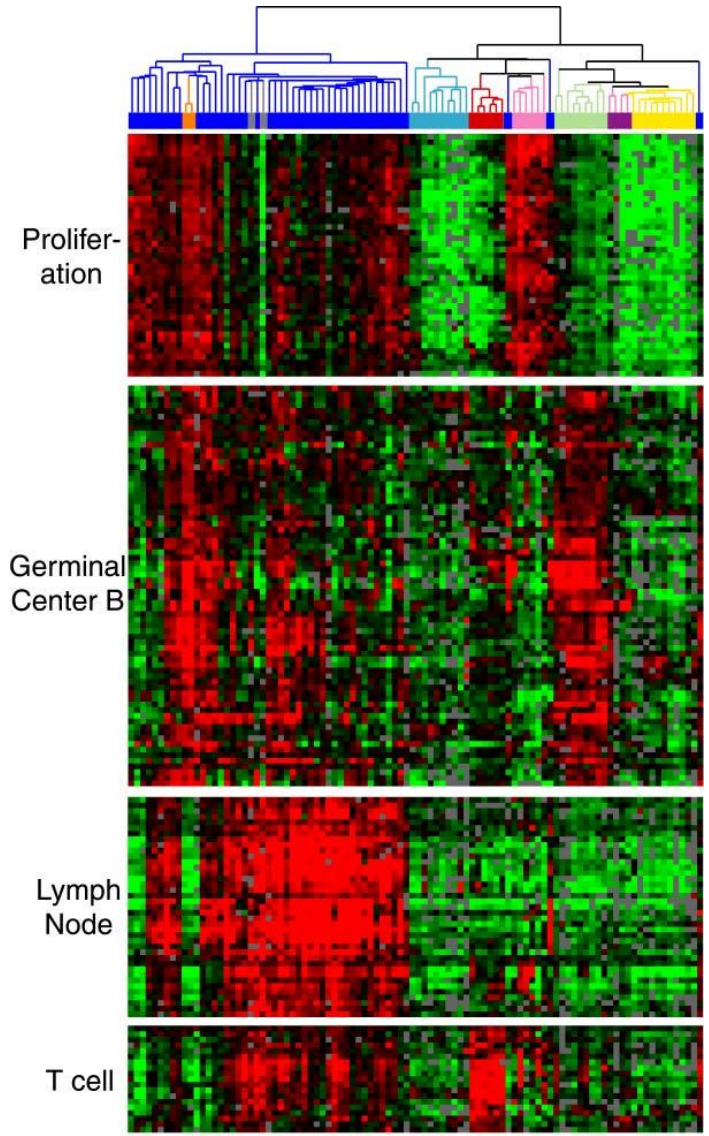
Problems with Hierarchical Clustering

- It is a global clustering algorithm.
- Considers *all* genes to be equally important for all samples.
- What if only a subset of the genes are co-expressed across only a subset of the samples?
- What if different subsets of the genes are co-expressed for different subsets of samples?

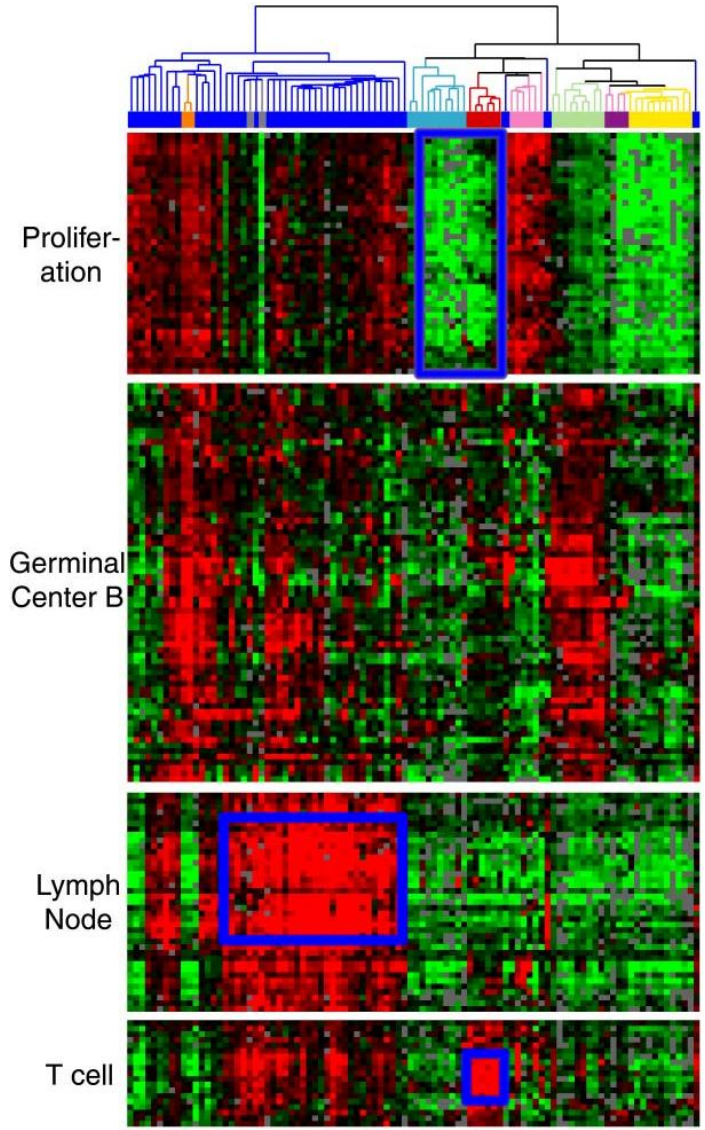
Example: Roberts et al. (Science 2000)



Example: Alizadeh et al. (Nature 2000)



Example: Alizadeh et al. (Nature 2000)



Biclustering

- A *bicluster* is a subset of genes and a subset of samples with the property that the selected genes are co-expressed only in the selected samples.

Biclustering

- A *bicluster* is a subset of genes and a subset of samples with the property that the selected genes are co-expressed only in the selected samples.
- By selecting samples *and* genes, a bicluster represents condition-specific patterns of expression.
- Issues in biclustering:

Biclustering

- A *bicluster* is a subset of genes and a subset of samples with the property that the selected genes are co-expressed only in the selected samples.
- By selecting samples *and* genes, a bicluster represents condition-specific patterns of expression.
- Issues in biclustering:
  - How do we measure the degree of co-expression of a subset of genes in a subset of samples?
  - How many biclusters should we compute?
  - How do we compare two different sets of biclusters?

History of Biclustering

- Block clustering: Hartigan 1972, recursively partition matrix into blocks.
- Biclustering formulated in the context of gene expression data by Cheng and Church, ISMB 2000.
- Since 2000, a number of papers have been published on biclustering.
  - Finding statistically-significant biclustering: Sharon, Tanay, and Shamir, ISMB 2002.
  - Iterative signature algorithm: Bergmann, Ihmels, and Barkai, Phys Review E 2003
  - Two surveys of biclustering:
    - Madeira and Oliveira, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2004.
    - Tanay, Sharan, and Shamir, Handbook of Computational Molecular Biology, 2006.

Biclustering: Cheng and Church

- Defined the score of a bicluster to be its mean squared residue.
- Developed an iterative algorithm for computing biclusters with residue less than δ (specified by the user) by addition and deletion of genes and samples.
- To find multiple biclusters, they "erase" the values in the previously-computed biclusters and continue.

Mean Squared Residue

- A = matrix of gene expression values, a_{ij} = value in the ith row and jth column of A.
- I = subset of genes/rows, J = subset of conditions/columns.
- A_{IJ} = submatrix of A containing the rows in I and the columns in J.
- The mean squared residue of A_{IJ} is

$$H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} - a_{IJ})^2 \text{, where}$$

- a_{iJ} = average of values in A_{IJ} along row i, a_{Ij} = average of values in A_{IJ} along column j and a_{IJ} = average of all values in A_{IJ}.

Examples of Mean Squared Residue

$$H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} - a_{IJ})^2$$

- Constant matrix:

Examples of Mean Squared Residue

$$H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} - a_{IJ})^2$$

- Constant matrix: 0.

Examples of Mean Squared Residue

$$H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} - a_{IJ})^2$$

- Constant matrix: 0.
- Single element:

Examples of Mean Squared Residue

$$H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} - a_{IJ})^2$$

- Constant matrix: 0.
- Single element: 0.

Examples of Mean Squared Residue

$$H_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{Ij} - a_{IJ})^2$$

- Constant matrix: 0.
- Single element: 0.
- Matrix with elements chosen randomly from the interval $[a, b]$ has expected mean squared residue $(b - a)^2/12$.

Computational Problem

- Given a matrix A and a parameter δ find the largest submatrix of A with mean squared residue less than δ.

Computational Problem

- Given a matrix A and a parameter δ find the largest submatrix of A with mean squared residue less than δ.
- Why find the largest submatrix?

Computational Problem

- Given a matrix A and a parameter δ find the largest submatrix of A with mean squared residue less than δ.
- Why find the largest submatrix?
- How do we measure size of a submatrix?

Computational Problem

- Given a matrix A and a parameter δ find the largest submatrix of A with mean squared residue less than δ.
- Why find the largest submatrix?
- How do we measure size of a submatrix?
  - Perimeter: maximise $|I| + |J|$.
  - Square: maximise $|I| = |J|$.
  - Area: maximise $|I||J|$.

Computational Problem

- Given a matrix A and a parameter δ find the largest submatrix of A with mean squared residue less than δ.
- Why find the largest submatrix?
- How do we measure size of a submatrix?
  - Perimeter: maximise $|I| + |J|$.
  - Square: maximise $|I| = |J|$.
  - Area: maximise $|I||J|$.
- How hard is the problem?

Computational Problem

- Given a matrix A and a parameter δ find the largest submatrix of A with mean squared residue less than δ.
- Why find the largest submatrix?
- How do we measure size of a submatrix?
  - Perimeter: maximise $|I| + |J|$.
  - Square: maximise $|I| = |J|$.
  - Area: maximise $|I||J|$.
- How hard is the problem?
  - Perimeter: maximise $|I| + |J|$, can be solved in polynomial time but is inappropriate when #rows $>>$ #columns.

Computational Problem

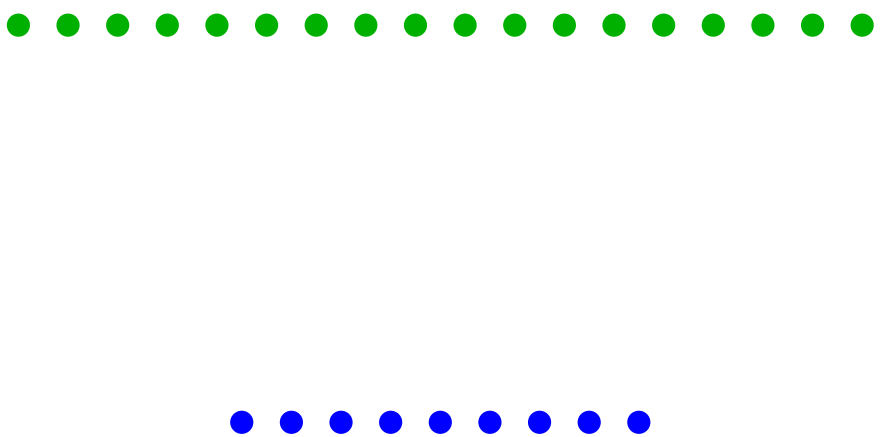
- Given a matrix A and a parameter δ find the largest submatrix of A with mean squared residue less than δ.
- Why find the largest submatrix?
- How do we measure size of a submatrix?
  - Perimeter: maximise $|I| + |J|$.
  - Square: maximise $|I| = |J|$.
  - Area: maximise $|I||J|$.
- How hard is the problem?
  - Perimeter: maximise $|I| + |J|$, can be solved in polynomial time but is inappropriate when $\#\text{rows} >> \#\text{columns}$.
  - Square: maximise $|I| = |J|$, NP-Hard .
  - Area: maximise $|I||J|$, also NP-Hard (proven after the Cheng and Church paper).

Algorithms

- Since the problems are computationally intractable, use heuristics to find biclusters of "large" size.
- Basic idea: add/delete a row/column until mean squared residue does not decrease.
  - Delete a row/column if its deletion improves mean squared residue.
  - Add a row/column if its addition improves mean squared residue.
  - Add some tricks to allow deletion/addition of multiple rows/columns so that it is not necessary to recompute mean squared residue after each change.

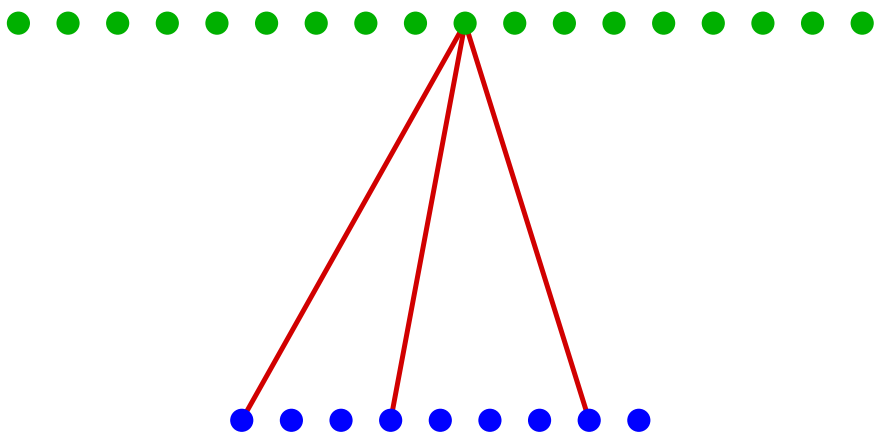
Biclustering: Sharan, Tanay, and Shamir

- Graph model: each gene is a node, each sample is a node.



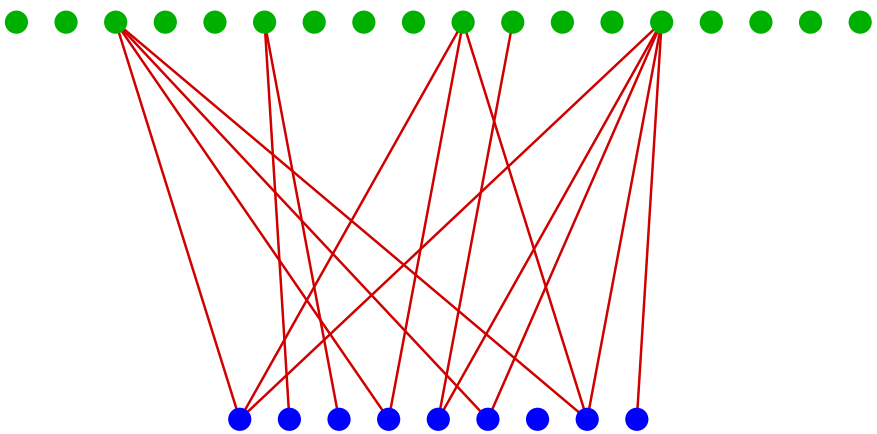
Biclustering: Sharan, Tanay, and Shamir

- Graph model: each gene is a node, each sample is a node.
- Bipartite graph: connect a gene to a sample if the gene *responds* in that sample, i.e., if the expression value is not between -1 and 1 after standardisation.



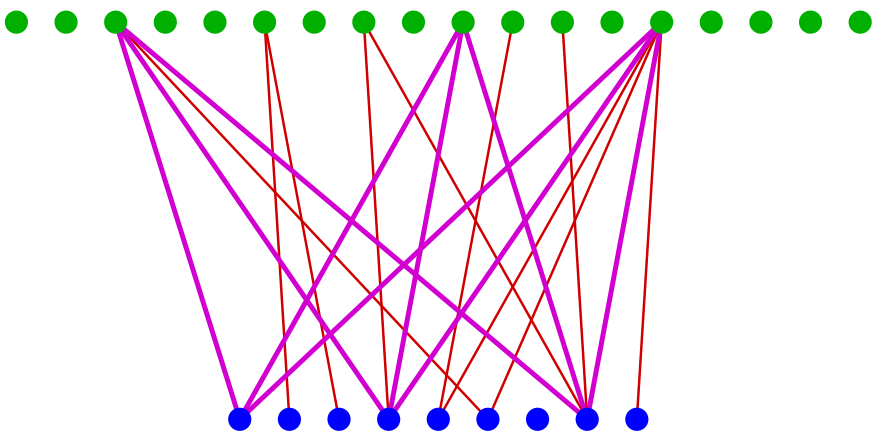
Biclustering: Sharan, Tanay, and Shamir

- Graph model: each gene is a node, each sample is a node.
- Bipartite graph: connect a gene to a sample if the gene *responds* in that sample, i.e., if the expression value is not between -1 and 1 after standardisation.



Biclustering: Sharan, Tanay, and Shamir

- Graph model: each gene is a node, each sample is a node.
- Bipartite graph: connect a gene to a sample if the gene *responds* in that sample, i.e., if the expression value is not between -1 and 1 after standardisation.
- Bicluster $\equiv$ clique; weight of a bicluster is the sum of the weights of its edges.



Computing Cliques

- Compute cliques with many edges or large weight.

Computing Cliques

- Compute cliques with many edges or large weight.
- Problem is *NP*-complete.

Computing Cliques

- Compute cliques with many edges or large weight.
- Problem is *NP*-complete.
- Assume each gene responds only in k samples.

Computing Cliques

- Compute cliques with many edges or large weight.
- Problem is *NP*-complete.
- Assume each gene responds only in k samples.
- Any clique has at most k samples in it.

Computing Cliques

- Compute cliques with many edges or large weight.
- Problem is *NP*-complete.
- Assume each gene responds only in k samples.
- Any clique has at most k samples in it.
- Algorithm:
  1. For every gene, consider all subsets of neighbouring samples.
  2. For each subset, find all the genes connected to all the samples in the subset.
  3. Keep track of the clique of largest weight.

Computing Cliques

- Compute cliques with many edges or large weight.
- Problem is *NP*-complete.
- Assume each gene responds only in k samples.
- Any clique has at most k samples in it.
- Algorithm:
  1. For every gene, consider all subsets of neighbouring samples.
  2. For each subset, find all the genes connected to all the samples in the subset.
  3. Keep track of the clique of largest weight.
- Running time is $O(n2^k)$, since each gene has at most $O(2^k)$ subsets of neighbouring samples. How do we obtain this running time? Exercise.

Computing Cliques

- Compute cliques with many edges or large weight.
- Problem is *NP*-complete.
- Assume each gene responds only in k samples.
- Any clique has at most k samples in it.
- Algorithm:
  1. For every gene, consider all subsets of neighbouring samples.
  2. For each subset, find all the genes connected to all the samples in the subset.
  3. Keep track of the clique of largest weight.
- Running time is $O(n2^k)$, since each gene has at most $O(2^k)$ subsets of neighbouring samples. How do we obtain this running time? Exercise.
- Add some heuristics: iteratively add/delete node that improves the weight the most until no modification is possible.

Assessing Edge Weights

- Goal: assess statistical significance of a bicluster in "random" data; assign edge weights so that weight of a bicluster is equal to its statistical signficance.
- U = set of genes, V = set of conditions, E is the set of edges.
- Simple model: assume each edge occurs with probability $p = |E|/(|U||V|)$.
- Given a bicluster $H = (U', V', E')$, the probability $p(H)$ of observing a bicluster at least as dense as H is the probability that if we select each of the $|U'||V'|$ possible edges with probability p, we will select E' or more edges:

Assessing Edge Weights

- Goal: assess statistical significance of a bicluster in "random" data; assign edge weights so that weight of a bicluster is equal to its statistical signficance.
- U = set of genes, V = set of conditions, E is the set of edges.
- Simple model: assume each edge occurs with probability $p = |E|/(|U||V|)$.
- Given a bicluster $H = (U', V', E')$, the probability $p(H)$ of observing a bicluster at least as dense as H is the probability that if we select each of the $|U'||V'|$ possible edges with probability p, we will select E' or more edges:

$$\sum_{|E'| \leq i \leq |U'||V'|} \binom{|U'||V'|}{i} p^i (1-p)^{|U'||V'|-i}.$$

Assessing Edge Weights Continued

- $p(H) = \sum_{|E'| \leq i \leq |U'||V'|} \binom{|U'||V'|}{i} p^i (1-p)^{|U'||V'|-i}$.
- If $p \leq 1/2$, $p(H) \leq 2^{|U'||V'|} p^{|E'|} (1-p)^{|U'||V'|-|E'|}$.
- To minimise $p(H)$, maximise $-\log p(H) = -|U'||V'| - |E'| \log p - (|U'||V'| - |E'|) \log(1-p)$.
- Assign each edge in the graph a positive weight $-1 - \log p$ and each edge not in the graph a negative weight $-1 - \log(1-p)$.